AD-A057 076 RAND CORP SANTA MONICA CALIF F/G 9/2
MODIA: VOLUME 3. OPERATION AND DESIGN OF THE USER INTERFACE - A--ETC(U)
APR 78 P CARPENTER-HUFFMAN, M FUJISAKI F49620-77-C-0023
UNCLASSIFIED RAND/R-1702/1-AF NL

| OF |
AD
AD57076

END
DATE
FILMED
9-78
DDC

LEVEL III

(11)

A040694

R-1702/1-AF
APRIL 1978

MODIA: Vol. 3 Operation and Design of the User Interface — Appendix

Polly Carpenter-Huffman, Misako Fujisaki, Ray Pyles

A Project AIR FORCE report
prepared for the
United States Air Force

Rand
SANTA MONICA, CA. 90406

The research reported here was sponsored by the Directorate of Operational Requirements, Deputy Chief of Staff/Research and Development, Hq. USAF under Contract F49620-77-C-0023. The United States Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon.

(9) Interim rept.,

(14) RAND/R-1702/1-AF

(11) APRIL 1978

(12) 44 p.

(6)

MODIA: Volume 3. Operation and Design of the User Interface — Appendix

(10) Polly/Carpenter-Huffman, Misako/Fujisaki, Ray/Pyles

A Project AIR FORCE report
prepared for the
United States Air Force

(15) F49620-77-C-0023

Rand
SANTA MONICA, CA. 90406

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER R-1702/1-AF	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) MODIA: Vol. 3 Operation and Design of the User Interface -- Appendix		5. TYPE OF REPORT & PERIOD COVERED Interim 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Polly Carpenter-Huffman, Misako Fujisaki, Ray Pyles		8. CONTRACT OR GRANT NUMBER(s) F49620-77-C-0023
9. PERFORMING ORGANIZATION NAME AND ADDRESS The Rand Corporation 1700 Main Street Santa Monica, Ca. 90406		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Project AIR FORCE Office (AF/RDQA) Directorate of Operational Requirements Hq USAF, Washington, DC 20330		12. REPORT DATE April 1978 13. NUMBER OF PAGES 38
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release; Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

No restrictions

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Air Force Training
Education
Planning
Military Training

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A detailed description of the User Interface computer program.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

PREFACE

This report documents research conducted under Project AIR FORCE by The Rand Corporation. The work described here was performed as part of the project entitled "Analysis of Systems for Air Force Education and Training" under Rand's Manpower, Personnel, and Training Program. It is an appendix to the third volume in a series presenting the MODIA planning system. MODIA, a *Method of Designing Instructional Alternatives*, is a system of people, computer programs, and procedures that allows the rapid specification and simulation of courses of instruction during the early stages of instructional design. It complements and augments the present Air Force Instructional System Development (ISD) process.

The development of MODIA has been supported by the Deputy Chief of Staff/ Personnel, Headquarters United States Air Force, and the Air Training Command, especially DCS/Technical Training, the Training Development Directorate, and personnel at the Keesler Technical School. It is part of Rand's continuing research effort in the areas of planning and management in education, education technology, and the cost and effectiveness of education systems.

This appendix to Vol. 3 is intended for those readers charged with maintaining and modifying the User Interface (UI) program. Aside from some additional design considerations not described in the text of the report, it contains only a more detailed description of the operation of the UI.

The series of MODIA reports includes:

R-1700-AF, *MODIA: Vol. 1, Overview of a Tool for Planning the Use of Air Force Training Resources*, Polly Carpenter-Huffman.

R-1701-AF, *MODIA: Vol. 2, Options for Course Design*, Polly Carpenter-Huffman.

R-1702-AF, *MODIA: Vol. 3, Operation and Design of the User Interface*, Polly Carpenter-Huffman, Misako Fujisaki, and Ray Pyles.

R-1703-AF, *MODIA: Vol. 4, The Resource Utilization Model*, Margaret Gallegos.

R-1704-AF, *MODIA: Vol. 5, A User's Guide to the Cost Model*, Ronald Hess and Phyllis Kantar.

CONTENTS

PREFACE iii

USER INTERFACE PROGRAM DESCRIPTION 1
Design Considerations 1
Program Description 3

USER INTERFACE PROGRAM DESCRIPTION

The following description of the User Interface (UI) program of MODIA (a *M*ethod *O*f *D*esigning *I*nstructional *A*lternatives) is intended for readers responsible for maintaining, modifying, or transporting that program to another computer system. It is technical in content, and we assume that the reader is familiar with program operation as described in the text of this report, R-1702-AF.

Figures and tables appearing in this appendix have been prefixed with an "A" (e.g., Table A.1, Fig. A.7). Numbers with no prefix refer to figures and tables in the text of the report.

Aside from some additional design considerations not described in the text, this appendix contains only a more detailed description of the operation of the UI.

DESIGN CONSIDERATIONS

When the UI was designed, it was unclear what computer the Air Training Command might select as the host for MODIA operations. One of the major goals of the design process, therefore, was to maximize the system's transportability. To this end, we made the following decisions:

- To use ANS FORTRAN as the language,
- To centralize I/O operations,
- To modularize individual program functions,
- To impose a sequential (jobstream or overlay) control structure.

The selection of ANS FORTRAN was a result of two considerations: (1) the greater transportability of the ANS subset of FORTRAN than of COBOL or PL/I, and (2) the slightly higher efficiency of FORTRAN in its use of core storage. Although FORTRAN is found on a variety of machines, each manufacturer has enhanced the ANS version to capitalize on the unique capabilities of his own machine. A program developer can be assured his program will operate on another machine only if the program is confined to the ANS subset of FORTRAN.

The selection of ANS FORTRAN as the implementation language constrained the design of the UI program in one very important way. Data files were limited to sequential formats. This limitation is significant because the UI program must converse interactively with a planner about a very complex problem. The constraints of a "reasonable" response time conflicted sharply with the time required to access several sequential files to obtain or save information of interest to the course design. To attain the desired response time, we found it necessary to retain a significant portion of the course design data in core memory during the operation of the UI program. This decision caused the amount of core memory required for UI operation to be higher than originally desired, particularly where resource constraints were described. This complex interactive program requires a fairly large amount of core storage during its operation—approximately 292 thousand eight-bit bytes of core memory on the IBM 370 computer for the largest module. As a result, the UI operation is limited to the medium-to-large scale computer

installations, including a few large "mini-computers," that have enough core storage available.

Input/Output (I/O) operations on different computer systems are frequently installation dependent. Even when the same main frame computer is used, the operating system and the characteristics of magnetic tapes, disks, and other recording media vary significantly from one computer installation to another. Although this problem is partly solved on most general-purpose computer systems by some logical/physical file assignment functions provided by the operating system for the recording media, interactive terminal I/O to an on-line planner is still unstandardized across computer systems. In most cases, even the interface to the routines that cause output and read input at the terminal varies from site to site. We therefore decided that the functional programs would communicate with the planner and the recording media that hold the intermediate course design information through internal UI routines. These routines, described in Table A.1, would be modified from site to site as the need arose. Modifications to the I/O operations between sites

Table A.1

INPUT/OUTPUT ROUTINE DESCRIPTION

Routine Name	Description
ITRALT	Controls and performs input/output operations to the planner's terminal.
IWRREC	Writes arrays of data to sequential files in a caller specified format.
IRDREC	Reads arrays of data from sequential files in a caller specified format.
IRDSMT	Controls the generation of the text of a question at a remote terminal (through ITRALT) and checks the reading and legality of the planner's response. Terminates UI program operation if the planner so requests.
ITOA	Converts an integer to its alphanumeric representation, one character per word.
ATOI	Converts an alphanumeric string that contains one alphanumeric character per word to an integer and indicates an error if a non-numeric character is encountered.
ISPACE	Spaces a specified number of lines at the remote terminal.
CONVRT	Converts an alphanumeric string to integer.
Additional routines may be required for the UI on computers other than the IBM 370. The following routines are currently required on the Honeywell computer at Keesler AFB:	
IBCDAS	Converts input BCD character string to ASCII from remote terminal.
IASBCD	Converts ASCII character string to BCD for output to remote terminal.

should be straightforward, although dependent upon the complexity of the change required. For example, a complex change requiring internal character conversion was carried out in only a few days on the Honeywell 6060 at Keesler AFB when MODIA was installed there.

To conserve core storage during operation, the UI program is broken into several functional modules, so that only the instructions and data relevant to a particular phase of the overall course planning process occupy core storage at any time. Without this modularity, the UI program would require several times the amount of core storage currently required. Each module communicates with the others by the sequential files already mentioned. The modularity also provides the basis for the iteration of the course design as described in the text.

Finally, because not all operating systems support complex job structures or program and data overlay operations, the control structure is confined to the simplest conceivable form. The UI program performs a sequential operation of the functional modules or planning phases, with the ability to start at any module in the sequence and to terminate at any point, permitting the planner to iterate the design and to abort a design session. Therefore, there are two different control structures—a job stream and an overlay. In the job stream structure, shown in Fig. A.1, each module operates as an independent program in sequence within a batch job stream in the computer system. The job stream checks a status word controlled by the UI program after the completion of each module to determine where control is to be transferred. Each module indicates by the status word whether the job is to be aborted or to continue to the next sequential module. The job is aborted if the planner enters a "Q" in response to any UI question.

In addition, the first module in the job stream permits the planner to select a planning phase. The selection is indicated to the job stream by the same status word, and this selection is performed by the job stream. In the overlay structure, a more innovative design can be envisioned, permitting the planner to cycle back to an earlier phase without terminating the job, but the only overlay structure used so far has simply mimicked the job stream structure described above.

The design of the UI program has been generally constrained by the effort to maximize transportability to other machines. Although other tradeoffs were considered, we decided the system's ultimate usefulness would be adversely affected if it were limited to a single computer or a limited subset of computers.

PROGRAM DESCRIPTION

The User Interface program consists of eight modules that produce intermediate descriptions of the course design for the planner and a ninth module that then generates the final course design data for the Resource Utilization Model (RUM):

1. Select the Planning Phase
2. Describe Objectives and Tests
3. Describe Student Population and Course Diversification
4. Describe Teaching Policy
5. Describe Test Details
6. Describe Resources
7. Describe Resource Constraints

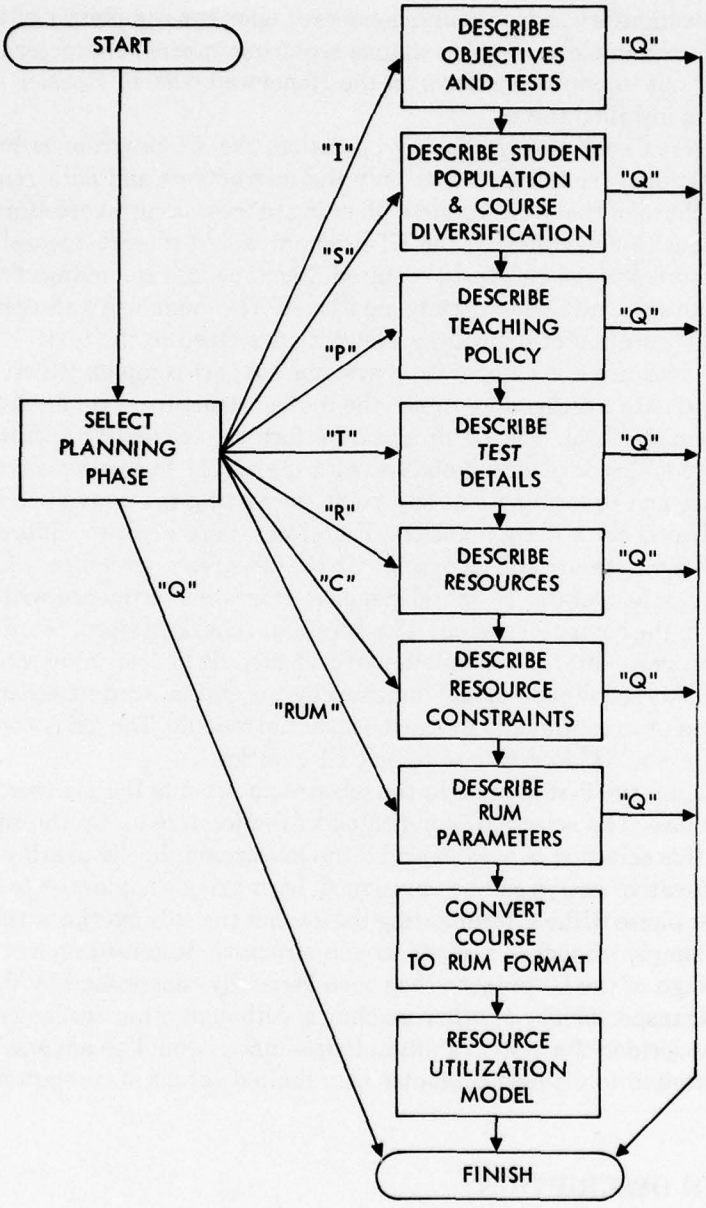


Fig. A.1—MODIA job stream control

8. Describe Resource Utilization Model Parameters
9. Convert Course Design Data to RUM Format

The following description includes flow diagrams for each module. The reader should refer to the text for examples of user interaction for the modules.

Before describing each module, we shall discuss their interaction by the sequential file media. Sequential disk files are used to maintain this communication, but tape files could be used with some minor modification of the system. The file names and the corresponding FORTRAN data set reference numbers are shown in Table A.2, and Table A.3 shows each file's use by each module. The files contain data about intermediate stages in the course design so that, by starting at the proper module, the planner can revise the course design without re-entering all data about the course.

Module 1: Select the Planning Phase

The first module allows the user to select his planning phase. If this is the planner's first session, the second module, *Describe Objectives and Tests*, must be selected; otherwise, any module may be selected for continuation or iteration of a course design. Fig. A.2 shows the flow diagram for this module; Fig. 5 illustrates a user interaction with it. Job stream control recognizes the selection and activities of the desired module.

Module 2: Describe Objectives and Tests

In this module, the planner lists the objectives to be taught and identifies the placement of tests and the related reviews and critiques. The flow diagram in Fig. A.3 shows the major steps taken in this module, and Fig. 6 gives an example of the interaction of the module with the planner.

Table A.2

USER INTERFACE FILES

FORTRAN Data Set Reference Number	File Name	For Format See
2	LEFIL.DESC	Fig. A.12
3	LEFIL.F24	Fig. A.11
4	ILEFIL	Fig. A.11
8	POP.FILE	Fig. A.6
9	OBJ.FILE	Fig. A.4
10	HMWKLN	Fig. A.8
11	SCHD.TIM	Fig. A.18
12	RUMINPU	Table A.9
13	RUMRES	Fig. A.16
14,2	LEFIL.D2 (File is referenced as 14 in Module 6; as 2 in Modules 7 and 9)	Fig. A.12

Table A.3

FILE USE

Module	LEFIL. DESC	LEFIL. F24	ILEFIL	POP. FILE	OBJ. FILE	HMWKLN	SCHD. TIM	RUMINPU	RUMRES	LEFIL. D2	Comments
Select the Planning Phase											No files created
Describe Objectives and Tests					0						
Describe Student Population and Course Diversification				0							
Describe Teaching Policy	0		0	I	I	0					
Describe Test Details	I	0	I	I/0	I						Data added to POP.FILE
Describe Resources	I	I			I					0	
Describe Resource Constraints		I/0			I	I			0	I	Data added to LEFIL.F24
Describe RUM Parameters				I/0			0				Student arrival characteristics replaced in POP.FILE
Convert Course to RUM Format		I		I	I		I	0	I	I	
RUM								I			Only reports created

I = Input
0 = Output

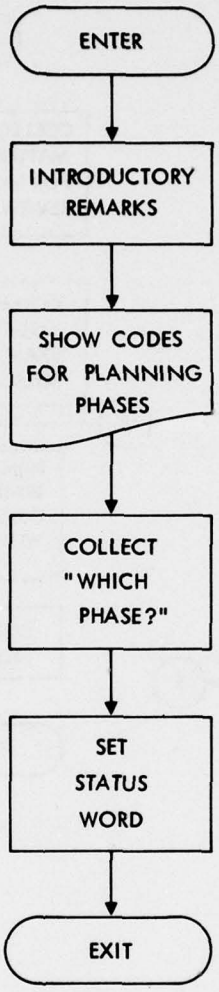


Fig. A.2—Select the planning phase flow diagram

First, the names of the objectives are collected in the order in which they will be taught. Next, each objective is presented to the planner in sequence, allowing him to assign one or more subject matter types appropriate for teaching that objective. Each objective is replicated by the UI program for each subject matter type assigned to it. This first expansion of the training objectives is printed. Then the program requests the tests, reviews, critiques, recycle points, and failure policies from the planner. If there are tests, the planner establishes the points at which they are given along with the subject matter types for each test and review. After the planner enters this information, the UI program presents the second expansion of the objectives, including the reviews, critiques, and tests in a resequenced list. This information is recorded in the file OBJ.FILE in the format shown in Fig. A.4.

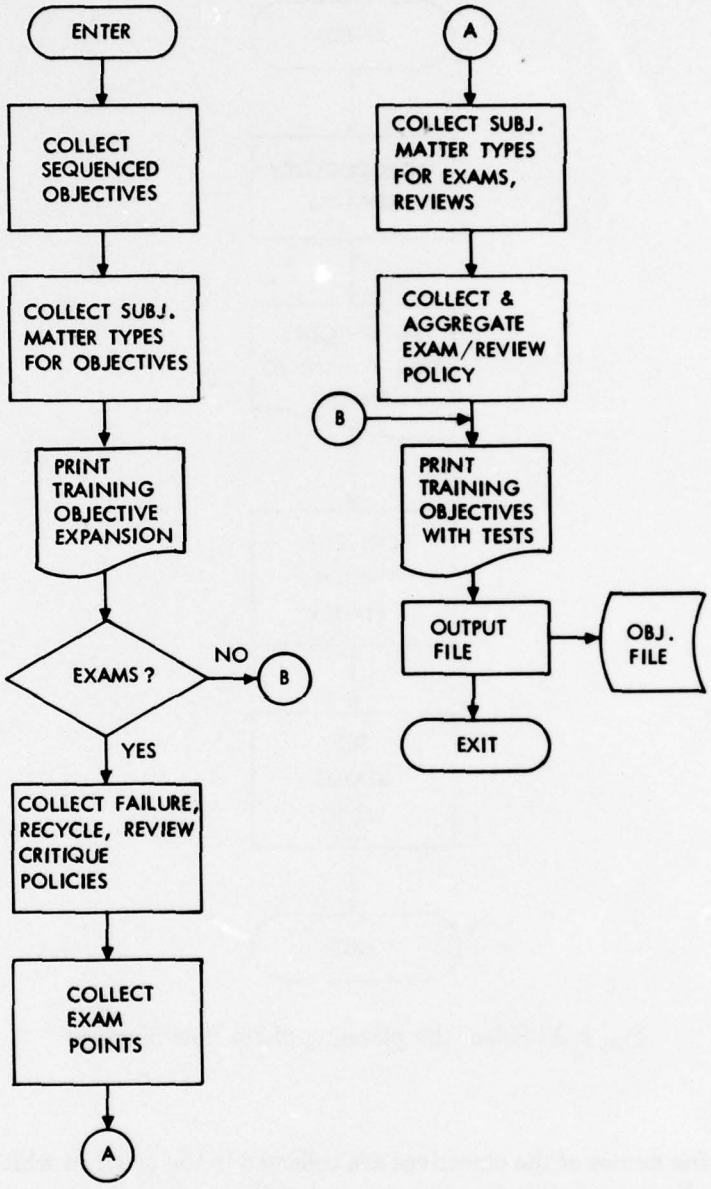


Fig. A.3—Describe objectives and tests flow diagram

KEY "1"	FAILRATE*	MAX.NO. RECYCLES*	N.OBJ*
	A5	A3	I3

Repeated N.OBJ times:

KEY "2"	OBJECTIVE NUMBER	OBJ.NAME*	S.M. TYPE	TEST
	A4	A8	A2	A1

KEY "3"	"0"
	I4

FAILRATE = Percentage of failures in the course.
MAX.NO.RECYCLES = Maximum number of recycles allowed on one test.
N.OBJ = Number of objective (review, critique) records.
OBJECTIVE NUMBER = Number of the objective corresponding to that record.
OBJ.NAME = User given name of objective, or "EXAMX," REVUEX," "CRITQX"; X indicates sequential number.
S.M.TYPE = Subject matter type.
TEST = Test indicator: "R" = Review; "S" = Exam, but another exam follows immediately or the test has a critique; "T" = Lone exam or last in a contiguous series; "C" = Critique.

*Table A.9 contains further details on these variables.

Fig. A.4—Training Objectives File: OBJ.FILE

Module 3: Describe Student Population and Course Diversification

In this phase, the planner describes how students arrive for the course and establishes whether the course is diversified and, if so, the forms of diversification and how the students are categorized.

The flow diagram for this module is shown in Fig. A.5, and Fig. 7 presents an example of the interaction of the planner with the module. The planner indicates the student arrival characteristics, including the arriving group size and frequency parameters, and decides whether a stochastic or deterministic arrival rate or group size will be simulated. The program then requests the planner to give the course diversification policy, indicating the existence and type of diversification by content, method, or tracking.

The student population categorization is collected. Here, the planner can divide the student population in three categories on the basis of: (1) ability alone, (2) "other characteristic," or (3) both "other characteristic" and the student's ability.

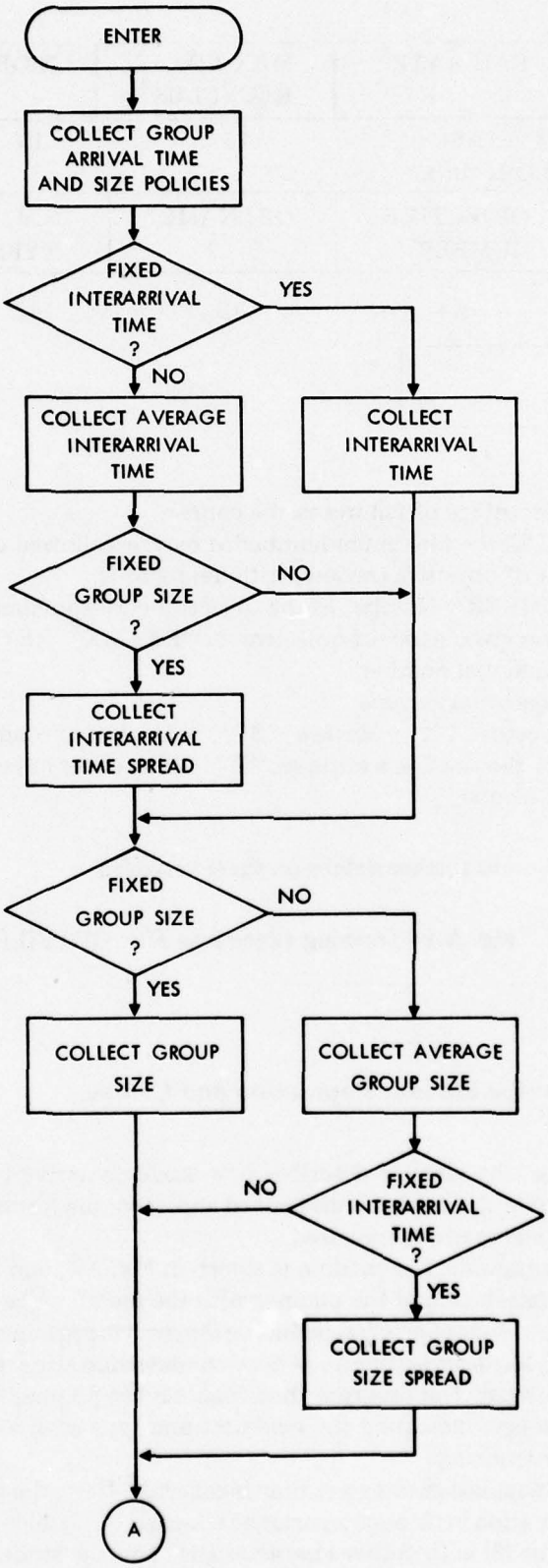


Fig. A.5—Describe student population and course diversification policy flow diagram (page 1 of 2)

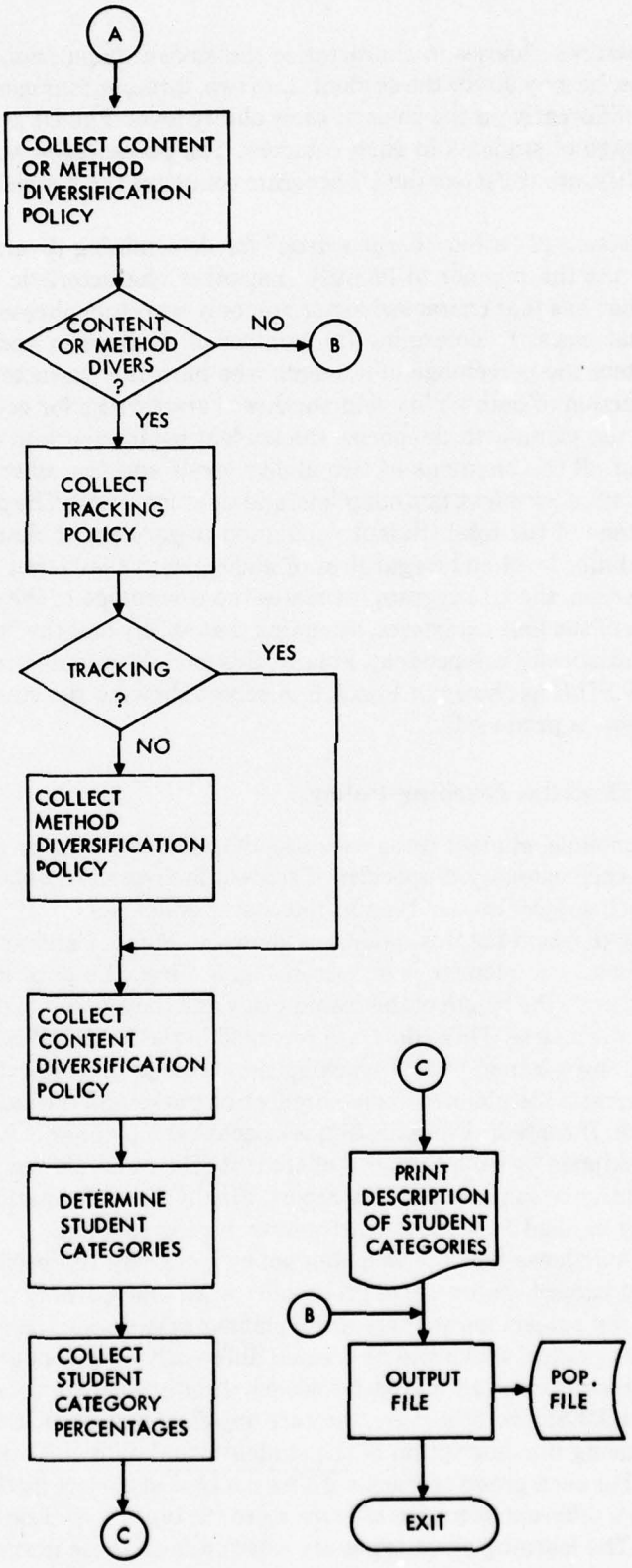


Fig. A.5—Continued (page 2 of 2)

If the planner chooses to characterize the student population on the basis of ability alone, he may divide the students into two, three, or four categories that will be treated differently on the basis of their ability level. The UI program requests the percentage of students in each category. The percentages, starting with the slowest ability, are collected; the UI program computes the percentage for the last category.

The selection of "other characteristic" for determining diversification of the course permits the planner to identify that other characteristic by name. Each student either has that characteristic or not; only two student categories are identified on that basis. To determine the number of students in each category, the planner enters the percentage of students who have the characteristic.

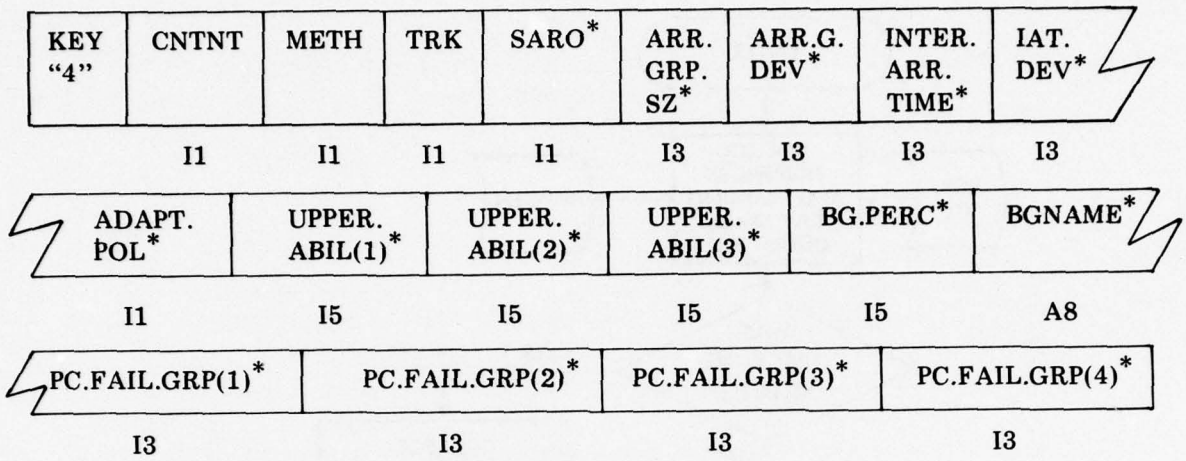
The selection of both ability and another characteristic for course diversification allows the planner to categorize the student population into four categories, consisting of all combinations of two ability levels and the other characteristics (e.g., fast pilots, slow pilots, fast nonpilots, and slow nonpilots). The planner specifies the percentage of the total student population (regardless of characteristic) with the lower ability level and (regardless of ability) with the "other characteristic." From these data, the UI program estimates the percentage of the student population in each of the four categories, assuming that ability and the "other characteristic" are statistically independent. Finally, this module outputs the information to the file POP.FILE as shown in Fig. A.6. A report showing the student population categorization is produced.

Module 4: Describe Teaching Policy

In this module, student categories are allocated to tracks or groups. Content skipped by each category is specified if content is diversified and the method for teaching each subject matter type in the course is defined.

The flow diagram for this module is shown in Fig. A.7 and an example of its interaction with the planner is shown in Fig. 8. First, the program requests the planner to specify the length of the training day and the amount of daily homework required in the course. These data are recorded in the file HMWKLN as shown in Fig. A.8. If the planner selects tracking as a course diversification policy, the program requests the planner to give number of tracks and the student categories in each track. If content diversification is selected, the planner indicates the objectives to be skipped by each category of student. The resultant content diversification information is summarized in a report. Finally, the information to determine the teaching method for each subject matter type is collected.

Figure A.9 shows that the teaching policy hierarchy represents the relationships among subject matter type, group (or track), and learning event types. For each of the ten subject matter types, the planner may divide the students into as many as four groups, which will be treated differently even though they have the same objectives. If tracking has been selected, these groups are the tracks specified in the LEFIL.DESC file; otherwise, they are any combination of student categories identified during the description of the student population and course diversification policy. For each group or track within each type of subject matter, the planner may choose a different sequence of learning event types to expose the material to be learned. The learning event types are selected from those permissible for each



CNTNT = Content diversification flag.
METH = Method diversification flag.
TRK = Tracking flag.
SARO = Student arrival rule option.
ARR.GRP.SZ = Student arrival group size.
ARR.G.DEV = Student arrival group size maximum spread.
INTER.ARR.TIME = Hours between student group arrivals.
IAT.DEV. = Interarrival time deviation.
ADAPT.POL = Student population breakdown key.
UPPER.ABIL (1), (2), (3) = Student ability level breakdown percentages.
BG.PERC = Percentage of students with other characteristic.
BGNAME = Name of other characteristic.
PC.FAIL.GRP(1), (2), (3), (4) = Percentages of failures from each student category (this information is added in the *Describe Test Details* module).

*Table A.9 contains further details on these variables.

Fig. A.6—Student Population and Course Diversification File: POP.FILE

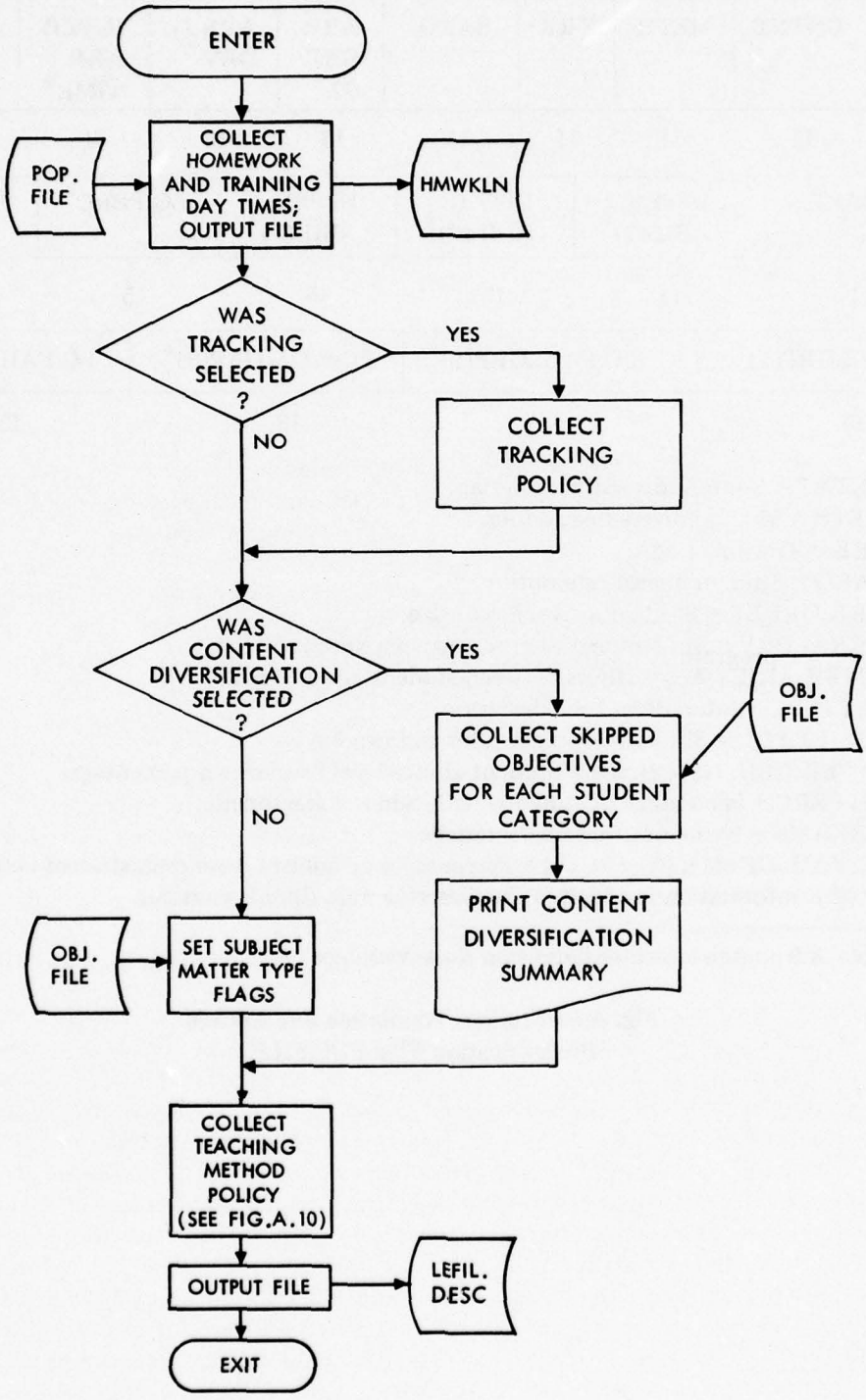


Fig. A.7—Describe teaching policy flow diagram

KEY	LEN	AVG
"b"	TRAINING	DAILY
	DAY	HMWK
	A3	A3

b = Blank
LEN TRAINING DAY = Average length of training day in minutes.
AVG DAILY HMWK = Average day's homework in minutes.

Fig. A.8—Homework and Training Time File: HMWKLN

subject matter type as shown in Table A.4. The permissible teaching format for each learning event type is shown in Table A.5, and the permissible teaching agent for each teaching format and subject matter type is shown in Table A.6.

The Teaching Policy routine flow diagram is shown in Fig. A.10, and an example of the user interaction begins on p. 3 of Fig. 8. The various shorthand codes for the teaching policy are printed first at the planner's option as shown on p. 2 of Fig. 8. Then, the program requests the planner to provide the teaching policy for each subject matter type appearing in the course. The planner's work is minimized in the following manner:

1. The planner may specify that a policy selected for a previous subject matter is to be also used for the present subject matter type;
2. The track or group may have a policy identical to one previously specified for a track or group within the same subject matter type;
3. Teaching method, teaching format, and teaching agent are automatically assigned when only a single option is available;
4. The planner is prompted with a list of allowable replies to each question.

Once these data are collected from the planner, they are applied to the list of objectives to produce the list of learning events for the course. The first action in this process is to expand the list of objectives into the appropriate number of tracks or groups if method diversification is selected and to provide a set of "flow codes" to reflect the student categories in each track or group. These flow codes consist of four-bit binary patterns of "1" or "0" to identify which student categories may take a particular learning event (1 if yes, 0 if no). If tracking has been chosen, flow codes are set so that each track will take a separate set of learning events. For subject matter type with method diversification and tracking, these sets will be different; for subject matter type without method diversification (and tracking) these sets will be the same. For subject matter type with method diversification and without tracking, flow codes are set so that each student group will take a separate set of learning events. For subject matter type without method diversification (and without tracking), the flow codes are set so that all students will take the same set of learning events.

Each regular objective (i.e., all except reviews, critiques, and tests) is expanded into a sequence of learning events, one learning event for each learning event type

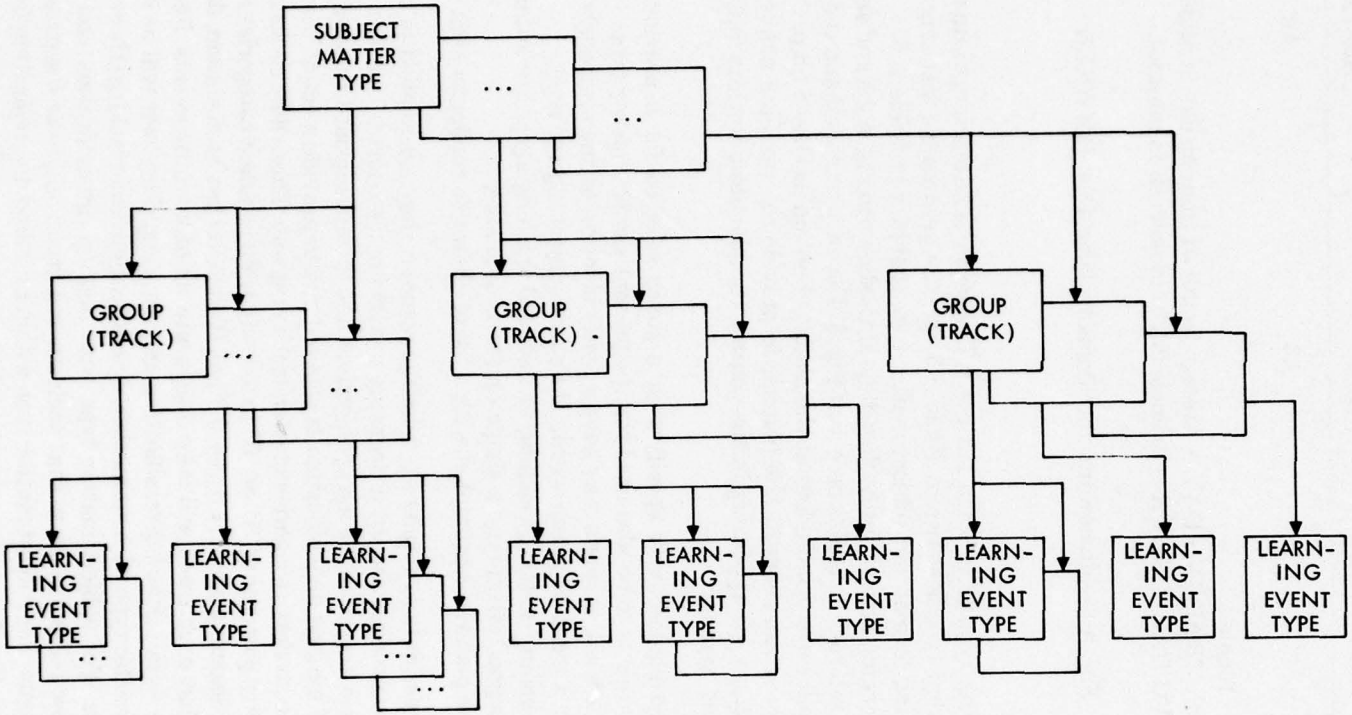


Fig. A.9—Teaching policy hierarchy

Table A.4

PERMISSIBLE TYPES OF LEARNING EVENTS FOR EACH SUBJECT MATTER TYPE

Subject Matter	Presentation/ Demonstration	Guided Practice	Unguided Practice	Group Discussion	Check Practice	Review	Test	Critique	Homework
1	X[a]			X	X	X	X	X	X
2	X[a]			X	X	X	X		X
3	X	X[a]	X	X	X	X	X		X
4	X	X[a]	X	X	X	X	X		X
5	X	X[a]	X	X	X	X	X		X
6	X	X[a]	X	X	X	X	X		X
7	X	X[a]		X	X	X	X		X
8	X	X[a]	X	X	X	X	X		X
9	X	X[a]	X	X	X	X	X		X
10	X	X[a]	X	X	X	X	X		X

[a]Event types most typical of instruction in subject matter type indicated.

Table A.5

PERMISSIBLE TEACHING FORMATS FOR EACH LEARNING EVENT TYPE

Learning Event Type	Teaching Format				
	Group Interaction	Simple	Recitation	Response-paced	Adaptive
Presentation/ Demonstration		X	X	X	X
Guided Practice		X	X	X	X
Unguided Practice		X	X		
Group Discussion	X				
Check Practice		X	X		
Review	X	X	X	X	X
Test		X	X		
Critique	X				
Homework		X	X	X	X

Table A.6

APPROPRIATE TEACHING AGENTS BY TEACHING FORMAT AND SUBJECT MATTER TYPE

Subject Matter Type	Teaching Format: Group Interaction	Simple	Recitation	Response-paced	Adaptive
1	I	L, I[a]	L, I[a]	L, I[a], RP	L, I[a], AP
2				L, I, RP	L, I, AP
3				L, I, RP	L, I, AP
4				L, I, RP[b]	L, I, AP[b]
5				L, I, RP	L, I, AP
6				L, I, RP[b]	L, I, AP[b]
7					
8					
9					
10					

L = Learner.
I = Instructor.
RP = Response-paced program.
AP = Adaptive program.

[a]Except homework.

[b]Presentation/demonstration only.

for each group eligible to take the objective. Reviews, critiques, and tests are converted on a one-for-one basis into learning events for each track or group. After the learning events are expanded, each learning event description is recorded in the file ILEFIL as shown in Fig. A.11. The complete list of learning events is printed for the planner. Administrative information is stored in the First Learning Event Description File LEFIL.DESC as shown in Fig. A.12.

Module 5: Describe Test Details

After the learning event sequence is established, the fifth module collects the detailed operations of the tests. The flow diagram for this module is shown in Fig. A.13, and an example of the interaction with the planner is shown in Fig. 13. Here the distribution of failures over the various student categories in the course is established along with the designation of which exams cause failures and recycles. As shown in Fig. A.11, these data, including the details for every learning event, are stored in file LEFIL.F24 with the extended record length. The failure rates for each student category are added to the POP.FILE.

Module 6: Describe Resources

In this module, the planner allocates resources to the learning events in the

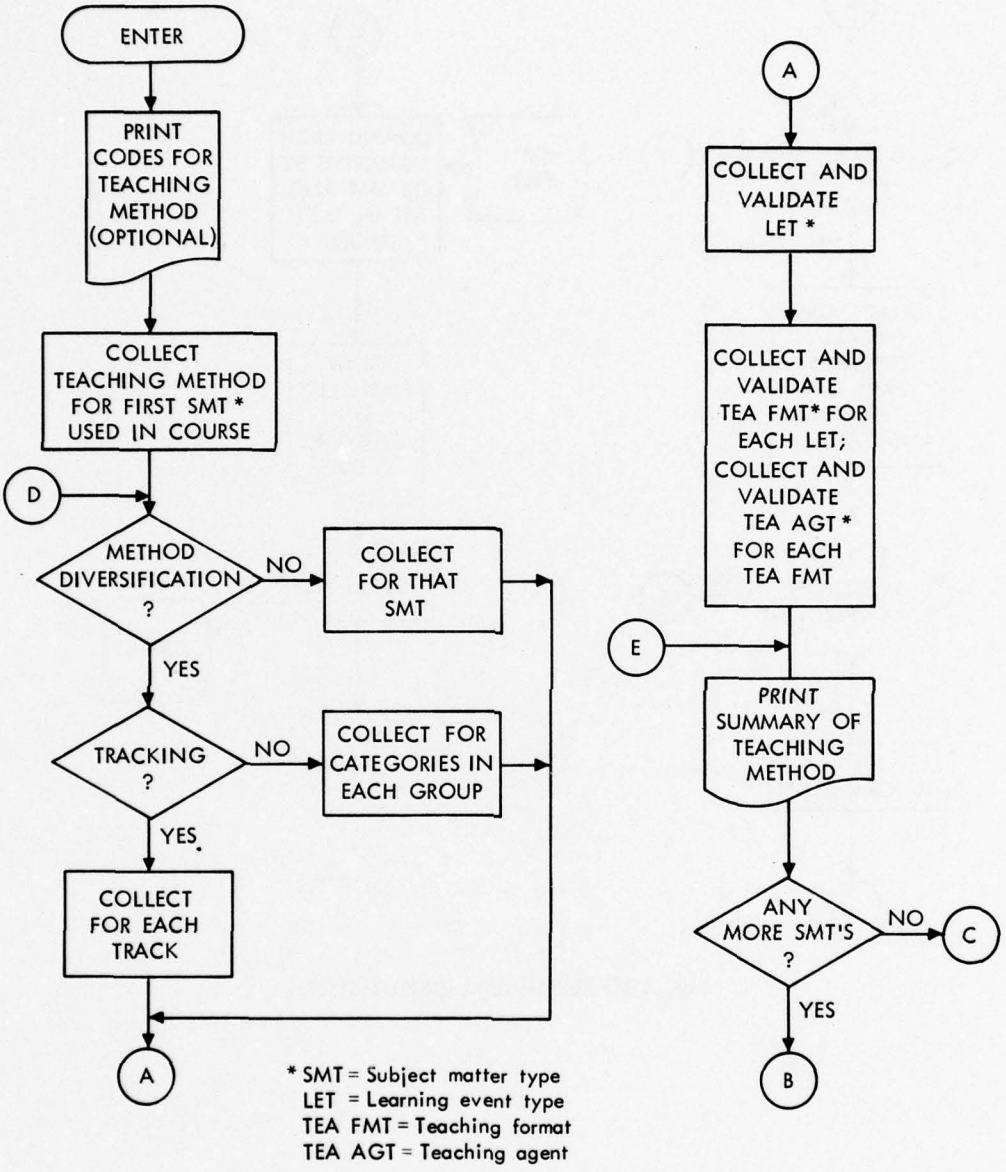


Fig. A.10—Teaching policy routine flow diagram (page 1 of 2)

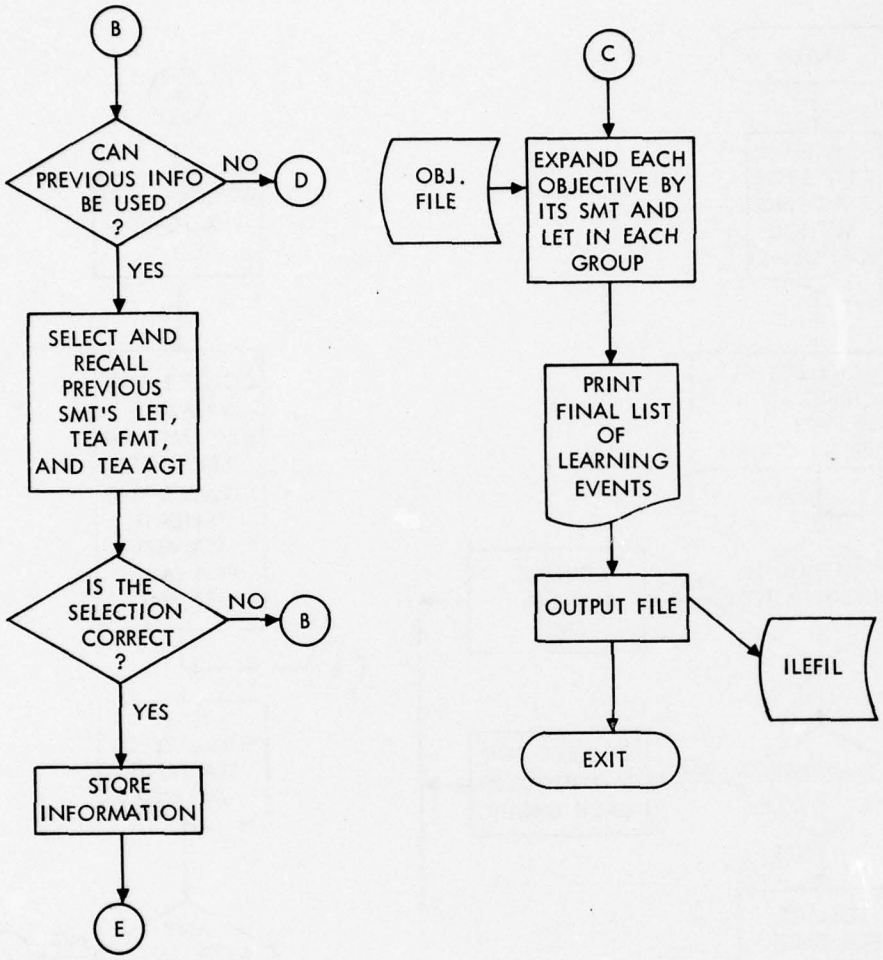


Fig. A.10—Continued (page 2 of 2)

KEY "b"	LENUMBER*	OBJ. NUM*	SUBJ. MATTER. TYPE*	EVENT. DESCR*	LE TYPE	FLOW. CODE*	TEST IND	TEA FMT	TEA AGT	GRP/ TRK NO.
	A3	I3	A2	A2	I1	I2	A1	I1	I1	I1

P.FAILS*	PC. RECYCLE*	LEREC*	MAX. STUDENTS. ALLWD*	MIN.NO. STUDENTS. REQD*	AVG. TIME. ALLWD*	MAX. TIME. MULT*	MIN. TIME. MULT*
I1	A3	A3	A6	A6	A6	A6	A6

For files ILEFIL and LEFIL.F24:

b = Blank.
LENUMBER = Learning event number.
OBJ.NUM = Objective name index.
SUBJ.MATTER.TYPE = Subject matter type.
EVENT.DESCR = Teaching method.
LE TYPE = Integer that points to teaching method literal.
FLOW.CODE = Code that indicates groups to take learning event.
TEST IND = Test indicator: 'R" = Review; "S" = Exam, but another exam follows immediately or the test has a critique; "T" = Lone exam or last in a contiguous series or a critique.
TEA FMT = Number that refers to teaching format.
TEA AGT = Number that refers to teaching agent.
GRP/TRK NO. = Number of group or track that is taking learning event (1 if no diversification).

The following are added for file LEFIL.F24:

P.FAILS = Failure key.
PC.RECYCLE = Percentage of students who will recycle back from test.
LEREC = Learning event for start of recycle.
MAX.STUDENTS.ALLWD = Maximum number of students allowed in learning event.
MIN.NO.STUDENTS.REQD = Minimum number of students allowed in learning event.
AVG.TIME.ALLWD = Completion time for learning event.
MAX.TIME.MULT = Maximum time multiple of AVG.TIME.ALLWD.
MIN.TIME.MULT = Minimum time multiple of AVG.TIME.ALLWD.

*Table A.9 contains further details on these variables.

Fig. A.11—Learning Event Files: *ILEFIL* and *LEFIL.F24*

KEY "þ"	N.LE DESCRIPTOR*	DIVS	TRKS	EXAMS	FAILS	RECYC	ADAPT	AP/RP
	I3	I1	I1	I1	I1	I1	I1	I1

INST	LEAR	EVAL	MONIT	OTHER CHARACTERISTIC NAME	NO. OF RES.
I1	I1	I1	I1	A8	I2

RESOURCE NAMES (31-8 Character names packed 62 characters/record in 4 records)
A62

RESOURCE ALLOCATION BIT MAP (31 bit information packed A4, 10/record in 25 records)
10 A4

POLICY EXISTENCE LIST-FIRST 513 ELEMENTS (Bit information packed I3, 20/record in 26 records)
20 I3

POLICY EXISTENCE LIST-ELEMENTS 514-553
40 I1

MEDIA EXISTENCE (for 31 resources)
31 I1

Fig. A.12—Learning Event Description Files:
LEFIL.DESC and LEFIL.D2 (page 1 of 2)

For files LEFIL.DESC and LEFIL.D2:

ƀ = Blank
N.LE.DESCRIPTOR = Total number of learning events.
DIVS = Diversification indicator.
TRKS = Number of tracks.
EXAMS = Exam flag.
FAILS = Failure flag.
RECYC = Recycle flag.
ADAPT = Adaptive policy.
AP/RP = Adaptive program or response-paced program as agent flag.
INST = Instructor as agent flag.
LEAR = Learner as agent flag.
EVAL = Evaluator flag.
MONIT = Monitor flag.
OTHER CHARACTERISTIC NAME = Other student characteristic name.
NO. OF RES. = Number of resources.
RESOURCE NAMES = User assigned resource names.
RESOURCE ALLOCATION BIT MAP = Resource allocation for each of the 250 learning events.
POLICY EXISTENCE LIST contains 553 elements as follows:

Elements	Contents
1-10	S.M. EXIST = Indicates if that subject matter exists
11-19	L.E.T. EXIST = Indicates if that learning event type exists.
20-23	TRACK = Indicates if that track exists.
24-113	S.M. L.E. TYPE = Indicates if that subject matter, learning event type combination exists.
114-473	S.M. TRK L.E. TYPE = Indicates if that subject matter, track and learning event type combination exists.
474-513	S.M. TRK/GRP = Indicates if that subject matter, track or group combination exists.
514-553	TRK/GRP CAT IND = Indicates if that track/group and subject matter combination exists.

The following is added for file LEFIL.D2:

MEDIA EXISTENCE = Indicator for resource used as MEDIA.

*Table A.9 contains further details on this variable.

Fig. A.12—Continued (page 2 of 2)

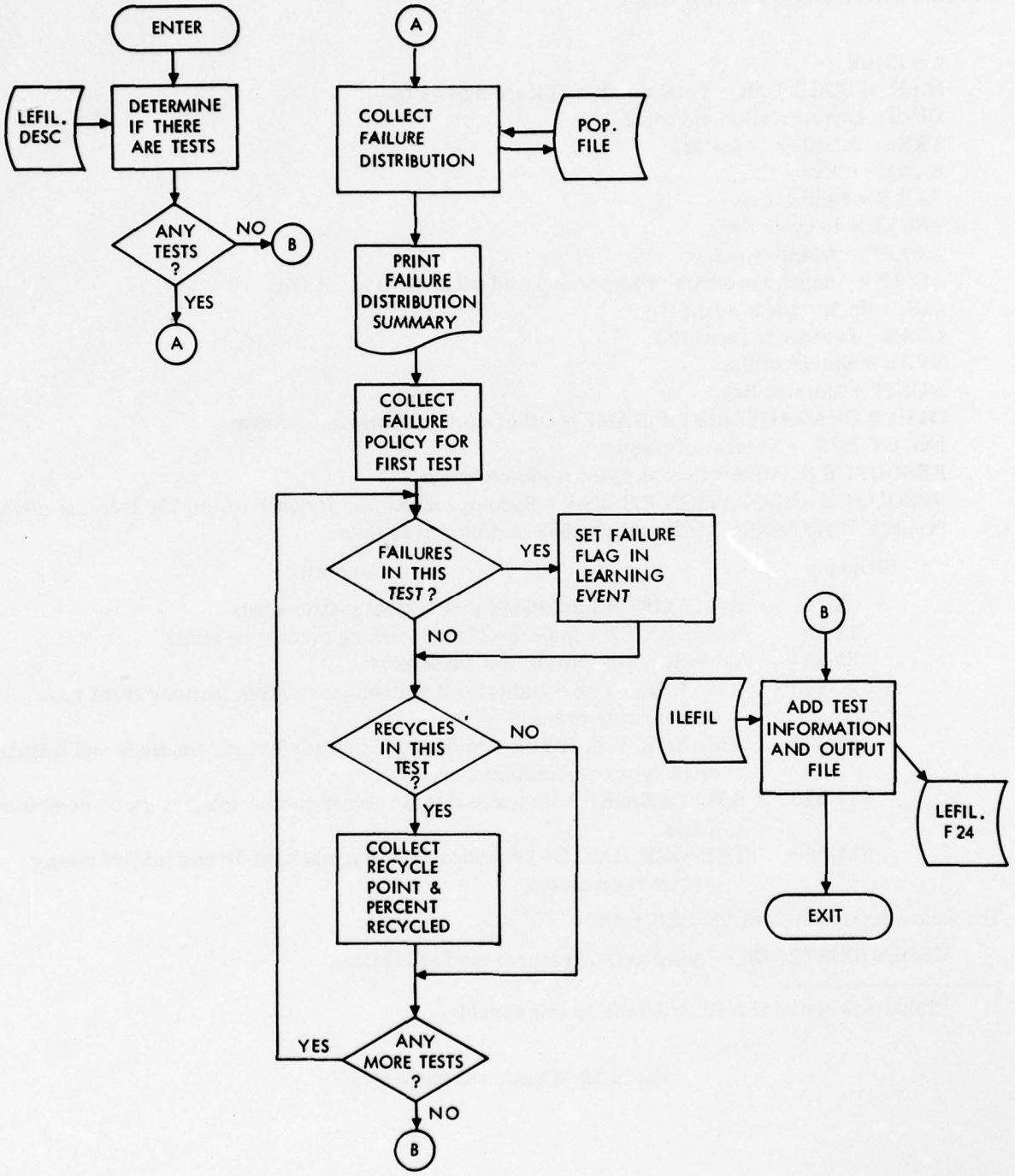


Fig. A.13—Describe test details flow diagram

course. The flow diagram is shown in Fig. A.14, and an example of an interaction with the planner is shown in Fig. 15.

First, the planner assigns special resources to learning events with subject matter types 7 through 10. The UI program prompts the planner for the assignments.

Next, the planner allocates the non-special resources. They are generated in the following resource category order: instructors, evaluators, monitors, facilities, instructor support media, learner support media, program support media, and recording hardware. The allocations are made learning event by learning event or by combinations of learning events. These combinations constitute the resource assignment policies. The combinations and their codes are shown in Table A.7. The UI program prompts the planner with acceptable resource assignment policies for each resource category; the valid assignment policies for the various categories are shown in Table A.8. In addition, for some resource categories, the program further validates the assignment (e.g., the program assigns instructor types only to those learning events for which the teaching agent is an instructor). See notes on Table A.8 for these validations.

The Total Resource Assignment Report is produced following the assignment. The planner is then given an opportunity to modify the resource assignment. The data are stored in file LEFIL.D2 in the format shown in Fig. A.12.

Module 7: Describe Resource Constraints

After all of the resources have been assigned, the seventh module, whose flow diagram is shown in Fig. A.15, helps the planner describe the resource constraints. Also in this phase, the planner determines the time and section size limitations of each learning event. An example of the planner interaction is shown in Figs. 19 through 26.

First, the UI program requests the capacity, quantity available, and sharing policy for each resource. Provision is included to accept:

1. Unknown capacities (to be estimated by RUM),
2. Capacities that vary from one learning event to another,
3. Unknown quantities (to be estimated by RUM),
4. Wholly dedicated resources,
5. Wholly shared resources,
6. A list of learning events that can share a resource.

The resource constraints (capacities, quantities available, and sharing policy) are recorded in the RUMRES file in the format shown in Fig. A.16. A summary of the resource constraints is produced at the planner's option.

Next, the program requests the quantities "maximum section size," "minimum section size," "average student time," "maximum allowed time factor," and "minimum allowed time factor" for all learning events. The planner may use the resource assignment policy codes to combine learning events, just as was possible for specification of the resources in Module 6. A report giving the "Complete Time and Section Size Assignment" is produced. The planner may correct the section size and time assignments at this point.

The planner then has the option of requesting the next two reports: "Summary of Course Design" and "Summary of Resource Characteristics." The "Summary of

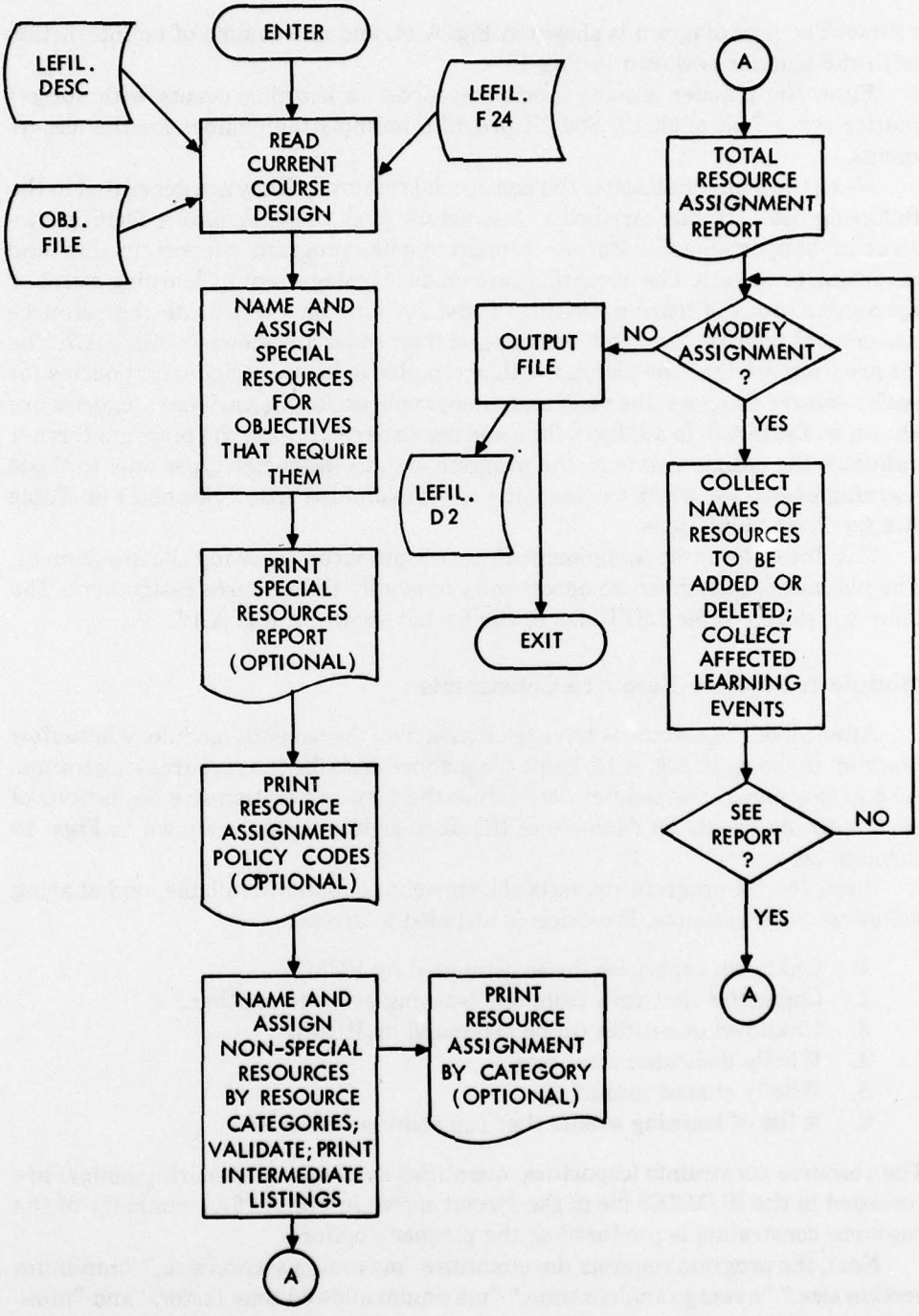


Fig. A.14—Describe resources flow diagram

Table A.7

LEARNING EVENT COMBINATIONS FOR RESOURCE ASSIGNMENT

Code	Meaning
N	None
W	Whole Course
B	Block of sequential learning events
SM	Subject matter type
T	Student track
LET	Learning event type
SMLE	Subject matter type and learning event type
SMGT	Subject matter type and student group/track
SMLS	Subject matter type, learning event type, and student group/track
LE	Individual learning event

Table A.8

VALID RESOURCE ASSIGNMENT POLICIES

Resource Category	Resource Assignment Policy Code									
	W	B	SM	LET	T	SMLE	SMGT	SMLS	LE	N
Special Resources									X	
Instructors[a]	X	X	X		X	X	X	X	X	
Monitors[b]	X	X	X		X	X	X	X	X	X
Evaluators[c]	X	X	X		X	X	X	X	X	X
Facilities	X	X	X		X	X	X	X	X	X
Instructor Support Media[a]	X	X	X	X	X	X	X	X	X	X
Learner Support Media[d]	X	X	X	X	X	X	X	X	X	X
Program Media[e]	X	X	X	X	X	X	X	X	X	X
Recording Hardware				X		X			X	X

[a]Instructors are assigned to learning events for which the teaching agent is an instructor.

[b]Monitors are assigned to learning events for which the subject matter type is 7 or greater, or the subject matter type is 6 or less and the teaching agent is not an instructor.

[c]Evaluators are assigned to learning events for which the learning event type is test, critique, or check practice.

[d]Assigned to learning events for which the teaching agent is a learner.

[e]Assigned to learning events for which the teaching agent is an adaptive program or a response-paced program.

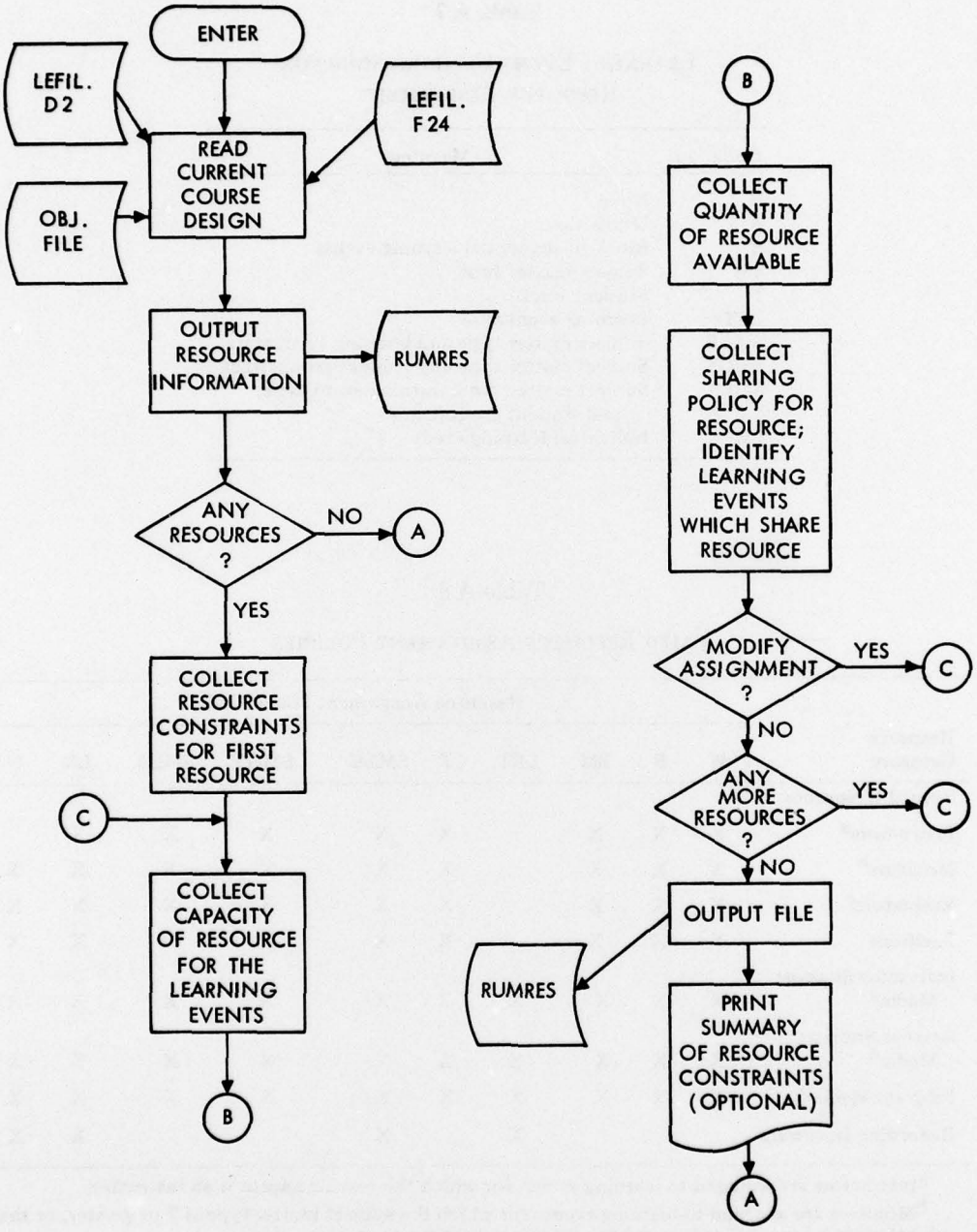


Fig. A.15—Describe resource constraints flow diagram (page 1 of 3)

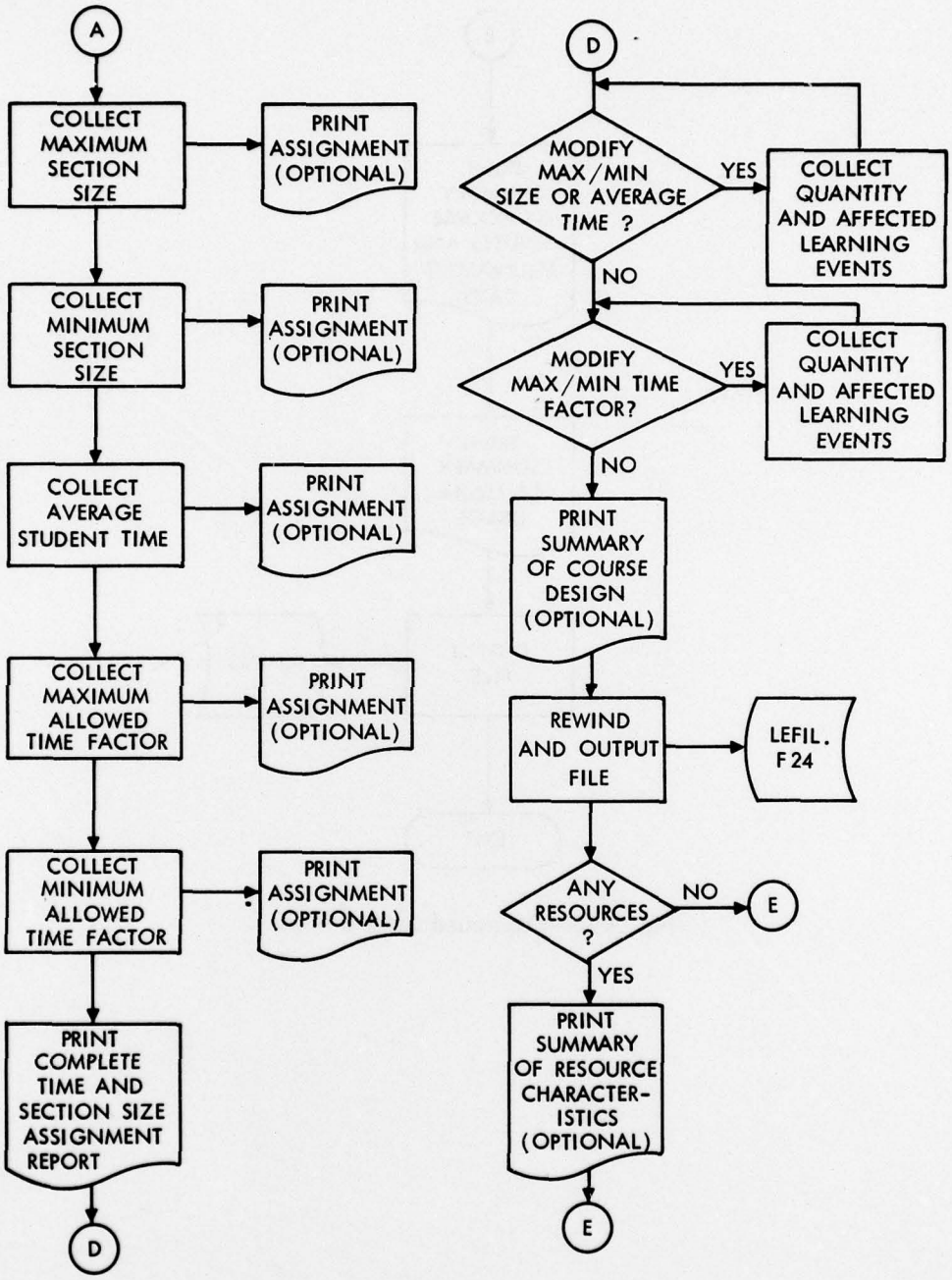


Fig. A.15—Continued (page 2 of 3)

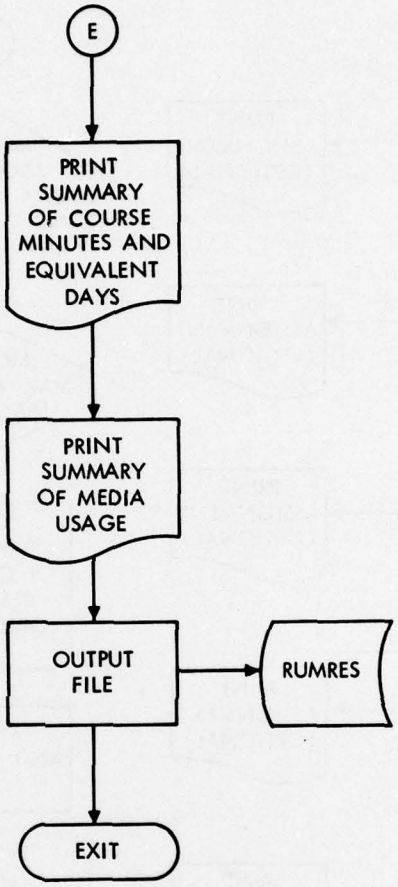


Fig. A.15—Continued (page 3 of 3)

31

KEY "♭"	N.RSOURCE.TYPE

I3

Repeated N.RSOURCE.TYPE times:

CAP.CODE	QTY.CODE	SHR.CODE

I7 I7 I1

KEY "♭"	NVAR.CAPS

I3

Repeated NVAR.CAPS times:

NVC (Packed 11 capacities per record for number of learning events)

11 I7

KEY "♭"	NVAR.SHR

I3

Repeated NVAR.SHR times:

KEY "♭"	NVS (Packed 31 items per record for number of learning events)

31 I1

Repeated 31 times:

NAME

A8

Repeated N.RSOURCE.TYPE times:

RESRC.REQMT.TABL (Packed 50 items per record for 250 learning events)

50 I1

Fig. A.16—Resource Constraints File: RUMRES (page 1 of 2)

b = Blank.
N.RSOURCE.TYPE = Number of resources.
CAP.CODE = Capacity code for the resource.
QTY.CODE = Quantity code of the resource.
SHR.CODE = Sharing code for the resource.
NVAR.CAPS = Number of resources with varying capacity.
NVC = Capacity of each resource with a varying capacity for the learning events.
NVAR.SHR = Number of variable sharing resources.
NVS = Share code of each resource with varying shareability for the learning events.
NAME = User assigned names for 31 resources.
RESRC.REQMT.TABL = Resource use indicator for 250 learning events.

Table A.9 contains further details on these variables.

Fig. A.16—Continued (page 2 of 2)

Course Design" is an extensive report and displays the learning events, the teaching method assigned to each, the categories of students taking each, the section size and time descriptions, and the resources assigned.

Two other reports are produced (not optional) in this phase: "Summary of Course Minutes and Equivalent Days" and "Summary of Media Usage." The first report summarizes the time spent in classroom instruction and in homework for each student category. These totals are printed both as minutes and as equivalent days using the length of the training day and the time required for daily homework. The "Summary of Media Usage" report displays the use of each media type in the course. The list is sorted by media type, objective, teaching agent, teaching format, learning event type, group or track, and subject matter type; it displays all learning events to which the planner has assigned media.

Module 8: Describe Resource Utilization Model Parameters

This module collects the remaining RUM parameters that control the length of the RUM simulation and the frequency with which the intermediate reports are generated. It also allows the planner to modify the student arrival characteristics. The flow diagram of the eighth module is shown in Fig. A.17; an example of the planner interaction is shown in Fig. 27.

After requesting the time interval between RUM reports, the UI program asks the planner to decide whether the number of graduates or the elapsed simulated time since the beginning of the course is to be used to terminate the simulation. In either case, the program then requests the value to be used to test for the end of the simulation. The run control data are recorded in the SCHD.TIM file in the format shown in Fig. A.18. The student population arrival rates and arrival group sizes may then be respecified by the planner as was done earlier in the *Describe Student Population and Course Diversification* module. The revised data are recorded in the file POP.FILE as shown in Fig. A.6, replacing those recorded earlier.

Module 9: Convert Course Design Data to RUM Format

The last module merges and transforms the UI data stored in the various intermediate files into a single file, RUMINPU, which represents the entire course design for input to the RUM. Some of the values stored as percentages in the intermediate files are converted to fractions. The order and format are shown in Table A.9. This is a straightforward mapping of available data from the existing UI files, so no flow diagram for this module is included here.

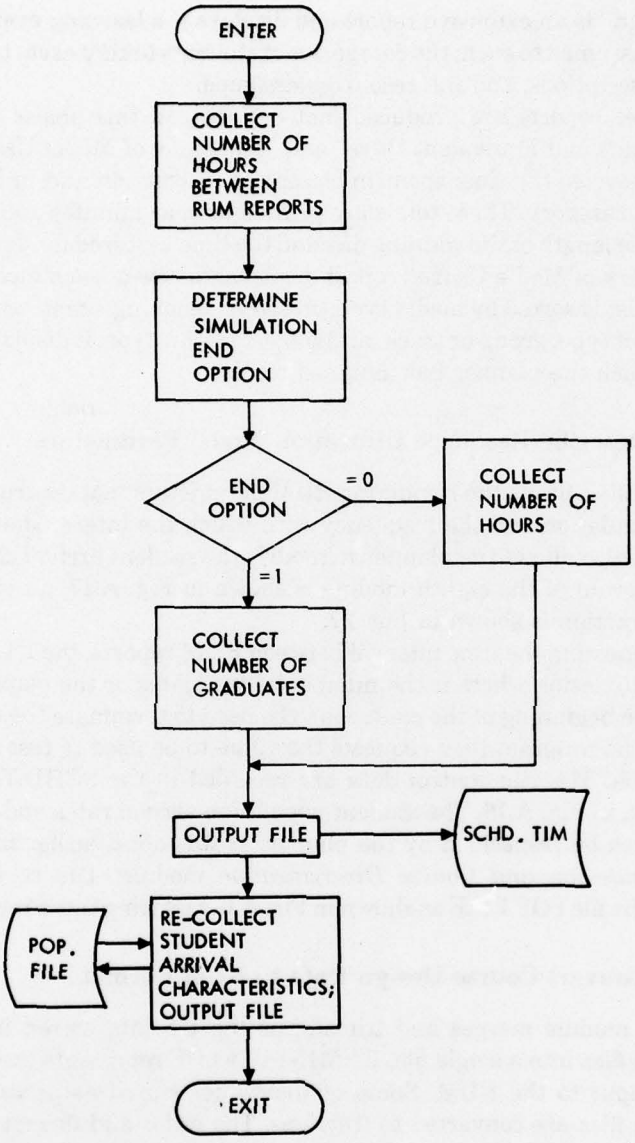


Fig. A.17—Describe resource utilization model parameters flow diagram

KEY "b"	REPORT INTERVAL	SIM. END. OPT	ENDVAR
	A4	A1	A4

b = Blank
REPORT.INTERVAL = Number of hours between RUM reports.
SIM.END.OPT = Simulation end option.
ENDVAR = End variable.

Table A.9 contains further details on these variables.

Fig. A.18—RUM Control File: SCHD.TIM

Table A.9

UI/RUM Interface File: RUMINPU

The UI program provides the RUM with the data described below in a "free-form," blank delimited, 80 character record file. The data items are described in the order in which they appear in the file RUMINPU, with the corresponding RUM variable names.

Name	Description	Format	Min	Max
SARO[a]	Student arrival rule option: 1: Fixed group size, fixed inter-arrival time. 2: Random group size, random inter-arrival time. 3: Random group size, fixed inter-arrival time. 4: Fixed group size, random inter-arrival time.	Integer	1	4
ARR.GRP.SZ	Group size or mean group size of arriving students.	Integer	1	999
INTER.ARR.TIME	Inter-arrival time or mean inter-arrival time of students.	Integer	1	999
ARR.G.DEV	"Three sigma" deviation from mean ARR.GRP.SZ above (provided if SARO is 3).	Integer	1	999
IAT.DEV	"Three sigma" deviation from mean INTER.ARR.TIME above (provided if SARO is 4).	Integer	1	999
REPORT.INTERVAL	Hours between RUM reports.	Integer	1	9999
SIM.END.OPT	Simulation end option; 0: run for ENDVAR simulated hours. 1: run until ENDVAR students graduate.	Integer	0	1
ENDVAR	Number of simulated hours or students graduated for RUM run.	Integer	1	9999
ADAPT.POL[b]	Course adaptability policy: 0: no adaptability 1: 2 groups based on some characteristic. 2: 2 groups based on ability 3: 3 groups based on ability 4: 4 groups based on ability 5: 4 groups based on ability and some other characteristic.	Integer	0	5
BGNAME	Name of the characteristic (provided if ADAPT.POL is 1 or 5).	Alpha (8 char.)	—	—
BG.PERC	Fraction of students with the other characteristic (provided if ADAPT.POL is 1 or 5).	Real	.01	.99
UPPR.ABIL(1)	Fraction of students in slowest ability group (provided if ADAPT.POL is from 2 to 5 inclusive).	Real	.01	.99
UPPR.ABIL(2)	Fraction of students in two slowest ability groups (provided if ADAPT.POL is 3 or 4).	Real	UPPR. ABIL(1)	.99

Table A.9—continued

Name	Description	Format	Min	Max
UPPR.ABIL(3)	Fraction of students in three slowest ability groups (provided if ADAPT. POL is 4).	Real	UPPR. ABIL (2)	.99
FAILRATE	Fraction of students who fail the course.	Real	0	.99
PC.FAIL.GRP	Fraction of students in each group who fail. Up to 4 values provided (depending on ADAPT.POL) in order of slowest ability group first and, within each ability, the group with the characteristic first.	Real Array	0	1
N.OBJ[c]	Number of objectives.	Integer	1	250
OBJ.NAME	Objective names (in order).	Alpha Array (8 char.)	—	—
N.LE.DESCRIPTOR	Number of learning events,	Integer	1	250
LENUMBER[d]	Learning event number.	Integer	1	N.LE. DES-CRIPTOR
OBJ.NUM	Related objective index.	Integer	1	N.OBJ
SUBJ.MATTER.TYPE	Subject matter type.	Integer	1	10
EVENT.DESCR	Teaching method.	Alpha (2 char.)	—	—
AVG.TIME.ALLWD	Average minutes for learning event.	Integer	0	999
FLOW.CODE	A 4 bit code indicating which students must attend the learning event. Starting with the least significant bit, the bits represent the slowest ability group and, within each ability, the group with the characteristic first.	Integer	1	15
MAX.STUDENTS.ALLWD	Maximum students for learning event.	Integer	MIN.NO. STUDENTS. REQD	999
MIN.NO.STUDENTS. REQD	Minimum students for learning event.	Integer	1	999
MAX.TIME.MULT	Maximum student rate (compared with average).	Real	1	9.90
MIN.TIME.MULT	Minimum student rate (compared with average).	Real	.01	1
P.FAILS	Failure key (provided if EVENT.DESCR is "T."): 0: no failures. 1: some failures.	Integer	0	1
PC.RECYCLE	Fraction of passing students recycling to earlier learning event (provided if EVENT.DESCR is "T.").	Real	0	1
LEREC	Learning event for start of recycle (provided if PC.RECYCLE greater than 0).	Integer	1	less than LENUMBER of the test
MAX.NO.RECYCLES	Maximum recycles allowed for a single student (provided if PC.RECYCLE greater than 0).	Integer	1	999

Table A.9—continued

Name	Description	Format	Min	Max
N.RSOURCE.TYPE[e]	Number of resources.	Integer	0	31
TYPE.ID.NO[f]	Resource index.	Integer	1	N. RSOURCE.TYPE
NAME	Resource name.	Alpha (8 char.)	—	—
CAP.CODE	Resource capacity code: -1: undefined capacity. 0: varying capacity (see NVC below). 1-999: number of students that one unit of the resource can accommodate.	Integer	-1	999
QTY.CODE	Resource quantity code: 0: unlimited quantity. 1-999: number of units of resource available.	Integer	0	999
SHR.CODE	Resource sharing code 1: dedicated to one learning event at a time. 2: may be shared by any number of learning events. 3: may be shared only by special learning events (see NVS below).	Integer	1	3
NVAR.CAPS[g]	Number of resources with varying capacity.	Integer	1	31
NVC	Capacities of variable capacity resources, provided in order of resources for each learning event.	Integer Array	1	999
NVAR.SHR[h]	Number of variable sharing resources.	Integer	1	31
NVS	Sharing codes (see SHR.CODE) for each resource in each learning event, provided in order of resources for each learning event.	Integer Array	1	2
RESRC.REQMT.TABL[i]	Resource allocation indicator, provided in order of resources for each learning event: 0: not needed. 1: needed.	Integer Array	0	1

[a]The data SARO through ENDVAR describe the student arrival characteristics and the control information for the RUM run that governs the frequency of reports and the duration of the simulation.

[b]The data ADAPT.POL through PC.FAIL.GRP describe the adapability of the course design to different students and the failure policy for the course.

[c]The data N.OBJ through N.LE.DESCRIPTOR describe the objectives in the course and indicate the number of learning events in the course.

[d]The data LENUMBER through MAX.NO.RECYCLES describe each learning event in detail and are recorded in order, one set for each learning event.

[e]N.RSOURCE.TYPE indicates the number of resources required for the course.

[f]The data TYPE.ID.NO through SHR.CODE describe each resource in turn and are recorded in order, one set for each resource.

[g]The data NVAR.CAPS through NVC describe the capacity of any resource whose capacity was specified as variable (CAP.CODE is zero). They are provided only for those resources if at least one variable capacity resource occurs.

[h]The data NVAR.SHR through NVS indicate the policy for variable sharing resources (SHR.CODE is 3) and are provided only for those resources if at least one resource with a variable sharing policy occurs.

[i]RESRC.REQMT.TABL represents the allocation of resources to learning events.